Exe 1, non-adaptive PRF (10 points).

A function family is a *non-adaptive* PRF, if it is a PRF according to the definition given in class, but its security should only holds against *non-adaptive* distinguishers: distinguishers that choose all queries to the oracle *before* making the first query (alternatively, they make all there queries at once).

1. Prove that an (adaptive) PRF is also a non-adaptive one
2. Assume OWF exists, prove there exist a non-adaptive PRF that is *not* an (adaptive) PRF

solution 1:1 Since every non-adaptive distinguisher is in particular a distinguisher, the result is immediate.

solution 1:2 Here is the idea: take a length preserving PRF and create a new ensemble as following: for each function f, declare the function $g_f(s) = f(s) \circ f(f(s))$. So g_f take a sting of length n and doubles it's length. In the additional n-bits g_f gives additional information that a regular distinguisher can use it and distinguish between such a function and random function. Such a distinguisher will just ask the oracle for $y' = g_f(0^n)$, and than ask again $y'' = g_f(y'_{1,\ldots,n})$. If the oracle is working with the above ensemble we expect that $y'_{n+1,\ldots,2n}$ will be equal $y''_{1,\ldots,n}$. A non-adaptive distinguisher won't be able do that, as we shell prove. Here comes the details:

Define first the following notation:
Given $\mathcal{P} = \{\mathcal{P}_n\}$ length preserving function ensemble ($\mathcal{P}_n$ contains functions that operating on n bits), Define the function ensemble $\mathcal{P}^{double} = \{\mathcal{P}_n^{double}\}$ as following:

$$\mathcal{P}_n^{double} = \{f^{double} : \{0,1\}^n \to \{0,1\}^{2n} \mid f^{double}(s) = f(s) \circ f(f(s)) \text{ where } f \in \mathcal{P}_n\}$$

So a function in $\mathcal{P}_n^{double}$ takes a function in $\mathcal{P}_n$ and let it act twice. First on the original input, and second on the generated output of the first step.

Assuming OWF exist we know that PRG exists. As we saw in class we know that there exist a PRF which is length preserving, denote it $\{\mathcal{P}_n\}$. We'll prove that $\{\mathcal{P}_n^{double}\}$ is a non-adaptive PRF, but isn't a (regular) PRF. First lets see:

Claim 0.1. *$\{\mathcal{P}_n^{double}\}$ isn't a PRF.*

Proof of Claim 0.1. Here is a simple oracle-aided PPT distinguisher, that distinguish between $\mathcal{P}_n^{double}$ and $\Pi_{n,2n}$. Denote by $Or()$ the oracle call.

Algorithm 0.2 (D: break $\mathcal{P}_n^{double}$)**.**

input*:* 1^n

- $y \leftarrow Or(0^n)$
- $z \leftarrow Or(y_{1,\ldots,n})$

- *if* $(y_{n+1,\ldots,2n} = z_{1,\ldots,n})$
- *return 1*
- *else*
- *return 0*

Obviously we have: $\mathsf{Pr}_{f\leftarrow\mathcal{P}_n^{double}}[D^f(1^n)=1]=1$. On the other hand: $\mathsf{Pr}_{f\leftarrow\Pi_{n,2n}}[D^f(1^n)=1]=\frac{1}{2^n}$, which proves that $\{\mathcal{P}_n^{double}\}$ isn't a PRF. □

Now we need to show that $\{\mathcal{P}_n^{double}\}$ is a non-adaptive PRF. Given an arbitrary oracle-aided PPT distinguisher D which is non-adaptive, we need to show that:

$$\left|\mathsf{Pr}_{f\leftarrow\mathcal{P}_n^{double}}[D^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\Pi_{n,2n}}[D^f(1^n)=1]\right|=neg(n) \tag{1}$$

Given such a distinguisher D, we create the following (possibly adaptive) distinguisher $\widetilde{D}(1^n)$ between $\mathcal{P}_n$ and Π_n:
$\widetilde{D}^f(1^n)$ just calls D. Each time D call its oracle with string s, $\widetilde{D}$ will call $f(s)$ than call $f(f(s))$, concatenate this 2 n-bits strings and return it to D. ($\widetilde{D}$ returns what D returns). More formally, for each ensemble $\mathcal{Q}_n$:

$$\widetilde{D}^{\mathcal{Q}_n}(1^n) \;=\; D^{\mathcal{Q}_n^{double}}(1^n)$$

Hence it's obvious that:

$$\mathsf{Pr}_{f\leftarrow\mathcal{Q}_n^{double}}[D^f(1^n)=1] \;=\; \mathsf{Pr}_{f\leftarrow\mathcal{Q}_n}[\widetilde{D}^f(1^n)=1] \tag{2}$$

Back to (1) we have:

$$\begin{aligned}
&\left|\mathsf{Pr}_{f\leftarrow\mathcal{P}_n^{double}}[D^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\Pi_{n,2n}}[D^f(1^n)=1]\right| \\
&\leq \left|\mathsf{Pr}_{f\leftarrow\mathcal{P}_n^{double}}[D^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[\widetilde{D}^f(1^n)=1]\right| && (3)\\
&+ \left|\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[\widetilde{D}^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\Pi_n}[\widetilde{D}^f(1^n)=1]\right| && (4)\\
&+ \left|\mathsf{Pr}_{f\leftarrow\Pi_n}[\widetilde{D}^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\Pi_n^{double}}[D^f(1^n)=1]\right| && (5)\\
&+ \left|\mathsf{Pr}_{f\leftarrow\Pi_n^{double}}[D^f(1^n)=1]-\mathsf{Pr}_{f\leftarrow\Pi_{n,2n}}[D^f(1^n)=1]\right| && (6)
\end{aligned}$$

According to (2) we get immediately that the value of the $|\ldots|$ in (3) and in (5) are 0. The value at (4) is negligible because we know $\mathcal{P}_n$ is PRF. So if we show that (6) is negligible we'll finish the work. The following claim will prove it:

Claim 0.3. Π_n^{double} *is computationally indistinguishable from* $\Pi_{n,2n}$ *for non-adaptive distinguishers.*

Proof of Claim 0.3. Suppose $D(1^n)$ is an oracle-aided PPT distinguisher. Denote by $p(n)$ the polynomial that bounds the running time of D. Denote by $x^1, x^2, \ldots, x^k$ the **different** n-bits strings that were delivered to the oracle by D, and $y^1, y^2, \ldots, y^k$ the $2n$-bits strings returned by the oracle. Define a *GOOD-RUN* of D, a run that satisfy the following: for each i, j, $1 \leq i, j \leq k$ $x^i \neq y^j_{1,\ldots,n}$. That mean we never got from the oracle a $2n$-bits string such that its first n bits were equal to one of our queries.

Lets consider an arbitrary *GOOD-RUN*. If D is working with an oracle randomly chosen from $\Pi_{n,2n}$, then all the $\{y^i\}_{i=1}^k$ are actually the same as if they were chosen from the uniform distribution U_{2n} (that is true even if it's not a *GOOD-RUN*). But this is the same whether D is working with an oracle randomly chosen from Π_n^{double}. The reason for this is the following:
Suppose the oracle is asked the i-th query: x^i. The first n-bits it returns are $\pi(x^i)$, where π was randomly chosen from Π_n. Since we assumed it was a *GOOD-RUN*, we know that the value of π on the i-th query is uniformly distributed on $\{0,1\}^n$. Hence the first n bits return by the oracle are the same if they were sample from U_n. What about the second half of y^i ? we know that they must be $\pi(\pi(x^i))$. But since it is a *GOOD-RUN*, we know that they also uniformly distributed on $\{0,1\}^n$, so those n bits are also the same as if they would be sampled from U_n.
To summarise, if we have a *GOOD-RUN* in both cases the run will be the same as if we were just sampling from the uniform distribution U_{2n} - those $2n$ bits. So in the situation of *GOOD-RUN*, those ensembles are statistically indistinguishable hence computationally indistinguishable.
What is the probability to get a *GOOD-RUN* ? We'll prove it is $1 - neg(n)$:
Since D is non-adaptive we can assume w.l.g that it chooses (not necessary randomly) different, $x^1, x^2, \ldots, x^k$, and than make k calls to the oracle, delivering to it the x^i one by one. Denote by A_i the event that y^i doesn't break the *GOODNESS* of the run, that is:

$$\forall j \neq i \quad y^i_{1,\ldots,n} \neq x^j$$

It's easy to see that if y^i is randomly chosen, then:

$$\mathsf{Pr}[A_i] = (1 - \frac{1}{2^n})^{k-1} \geq 1 - \frac{k-1}{2^n}$$

Hence:

$$\mathsf{Pr}[\overline{A_i}] \leq \frac{k-1}{2^n} < \frac{k}{2^n} \tag{7}$$

Obviously a run is *GOOD-RUN* iff $\cap_{i=1}^k A_i$. Now we'll give a lower bound on its probability. The following hold for a function randomly chosen from both ensembles.

Using (7) we get:

$$\begin{aligned}\mathsf{Pr}[\textit{GOOD-RUN}] &= \mathsf{Pr}[\cap_{i=1}^{k} A_i] = 1 - \mathsf{Pr}[\overline{\cap_{i=1}^{k} A_i}] \\ &= 1 - \mathsf{Pr}[\cup_{i=1}^{k} \overline{A_i}] \geq 1 - \sum_{i=1}^{k} \mathsf{Pr}[\overline{A_i}] \qquad \geq 1 - \sum_{i=1}^{k} \frac{k}{2^n} = 1 - \frac{k^2}{2^n}\end{aligned}$$

Since k is at most $p(n)$, we conclude that the probability NOT to get *GOOD-RUN* is negligible, hence the algorithm D will distinguish between the ensembles in negligible probability. □

Exe 2, Weak PRFs (10 points). A function family is a *weak* PRF, if it is a non-adaptive PRF according to the above definition, but its security should only holds against (non-adaptive) distinguishers who choose their queries *uniformly and independently* from the set of all possible queries.

1. Assume OWF exists, prove there exists a weak PRF that is not a non-adaptive PRF

solution 2 As in question 1, take a PRF $\{\mathcal{P}_n\}$ such that for each function $f \in \mathcal{P}_n$ $f : \{0,1\}^n \to \{0,1\}^n$.
Lets build the following new ensemble $\widetilde{\mathcal{P}}_n$ based on $\mathcal{P}_n$ as following. For each $f \in \mathcal{P}_n$, take $\widetilde{f}$ to be:

$$\widetilde{f}(s) = \begin{cases} f(s) & \text{if } s \neq 0^n \\ 0^n & \text{if } s = 0^n \end{cases}$$

So $\widetilde{f}$ is the same as f we just force that 0^n will go to 0^n.

Claim 0.4. *$\widetilde{\mathcal{P}}_n$ isn't non-adaptive PRF*

Proof of Claim 0.4. The following oracle-aided PPT distinguisher, distinguish between $\widetilde{\mathcal{P}}_n$ and Π_n, with non negligible probability. Denote by $Or()$ the oracle call.

Algorithm 0.5 (A: break $\widetilde{\mathcal{P}}_n$)**.**

input*:* 1^n

- *if* $(Or(0^n) = 0^n)$
- *return 1*
- *else*
- *return 0*

Obviously we have: $\mathsf{Pr}_{f \leftarrow \widetilde{\mathcal{P}}_n}[A^f(1^n) = 1] = 1$. On the other hand $\mathsf{Pr}_{f \leftarrow \Pi_n}[D^f(1^n) = 1] = \frac{1}{2^n}$, because this are the chances that 0^n will go to 0^n.
That proves that $\{\widetilde{\mathcal{P}}_n\}$ isn't a non-adaptive PRF. □

What left to prove is the following claim:

Claim 0.6. *$\{\widetilde{\mathcal{P}}_n\}$ is a weak PRF.*

Proof of Claim 0.6. Suppose $D(1^n)$ is an oracle-aided PPT distinguisher, that randomly choose the inputs to the oracle, aimed to break $\{\widetilde{\mathcal{P}}_n\}$. Similarly to question 1, define the notion of *GOOD-RUN* of D, but now as: a run that never asked the oracle the value for 0^n. It's easy to see that for both ensembles $\{\widetilde{\mathcal{P}}_n\}$, and $\{\mathcal{P}_n\}$, the

probability to get a *BAD-RUN* is negligible. (much similar calculation to question 1). We have:

$$\begin{aligned}
\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1] &= \mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}]\cdot\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[\textit{GOOD-RUN}] \\
&+ \mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{BAD-RUN}]\cdot\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[\textit{BAD-RUN}] \\
&= \mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}]\cdot(1-neg(n)) \\
&+ \mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{BAD-RUN}]\cdot(neg(n) \\
&= Pr_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}] + neg(n) \qquad (8)
\end{aligned}$$

Exactly the same computation will show us that:

$$\mathsf{Pr}_{f\leftarrow\widetilde{\mathcal{P}}_n}[D^f(1^n)=1] = Pr_{f\leftarrow\widetilde{\mathcal{P}}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}] + neg(n) \qquad (9)$$

Since when there is a *GOOD-RUN*, the run will be the same for both ensembles we get that:

$$Pr_{f\leftarrow\widetilde{\mathcal{P}}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}] = \mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1 \mid \textit{GOOD-RUN}] \qquad (10)$$

Combining (8), (9), and (10) we'll get:

$$\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1] = \mathsf{Pr}_{f\leftarrow\widetilde{\mathcal{P}}_n}[D^f(1^n)=1] + neg(n)$$

Hence if:

$$\left|\mathsf{Pr}_{f\leftarrow\widetilde{\mathcal{P}}_n}[D^f(1^n)=1] - Pr_{f\leftarrow\Pi_n}[D^f(1^n)=1]\right| \leq neg(n)$$

Then also:

$$\left|\mathsf{Pr}_{f\leftarrow\mathcal{P}_n}[D^f(1^n)=1] - Pr_{f\leftarrow\Pi_n}[D^f(1^n)=1]\right| \leq neg(n)$$

Since D was arbitrary, the proof is done. □

Exe 3. PRF to PRG. (10 points) Show that if there exist "not trivial" pseudorandom functions ensemble $\mathcal{F} = \{\mathcal{F}_n\}$ (i.e., the domain of $f \in \mathcal{F}_n$, is $\{0,1\}^{\ell(n)}$ for a polynomial-time computable $\ell(n) \in \omega(\log n)$), then there exist pseudorandom generators.

Note that there are no assumptions on the output length of the functions. Also don't go through one-way functions (unless you like to fully prove that one-way functions imply pseudorandom generators...)

solution 3 Before we give the formal proof, here is idea that stands behinds the construction. A PRG function is a construct that can take k random bits, and produce $k+i$ bits that seem to look like they are also random. So it's a mechanism that knows how to produce more random bits than originally given to it. In order to create such a mechanism from PRF we remember that:

- PRF get the randomness in the PPT $D(1^n)$ that randomly choose a function from the ensemble.
- The randomness generated by the PRF is actually the chosen function.
-

So what we need to do, is make sure that a PRF produce more randomness that it consumes. Here is the formal proof:
Before we start lets make precise some assumption about ensembles:
Suppose $\{\mathcal{F}_n\}$ is an effective function ensemble. As we defined it, there exist a poly time PPT algorithm $D(1^n)$, that return a description of a function in $\{\mathcal{F}_n\}$. Also the distribution of $\{\mathcal{F}_n\}$, is actually determined by $D(1^n)$. Since D is polynomial algorithm, we can assume that there is a polynomial $p(n)$ that bounds the number of random bits that $D(1^n)$ access. We can also refer to $D(1^n)$ as a deterministic algorithm that get a sequence of random bits $s \in \{0,1\}^{p(n)}$ and actually act as: $D(1^n, s)$. Note that D, need not use all the random bit it got, and on some situations use more random bits than other. So by definition we can assume that the distribution $\{\mathcal{F}_n\}$ is equivalent to:
Chosen uniformly $s \leftarrow U_{p(n)}$ then compute: $D(1^n, s)$.

We know that each function in $f \in \{\mathcal{F}_n\}$, is: $f : \{0,1\}^{l(n)} \to \{0,1\}^{m(n)}$. So the truth table of such f is actually a string of size $m(n) \cdot 2^{l(n)}$. We have:

$$l(n) \in \omega(log(n)) \quad \Rightarrow \quad l(n) \in \omega(log(p(n))) \quad \Rightarrow \quad 2^{l(n)} \in \omega(p(n))$$

Hence we can assume that for large enough n, we have $2^{l(n)} > (p(n))$. From now we concentrate on those n's. Define now a function g that we prove to be a PRG:

$$g : \{0,1\}^{p(n)} \to \{0,1\}^{m(n)\cdot(p(n)+1)} \text{ as } g(s) = D(1^n, s)(\overline{0}) \circ D(1^n, s)(\overline{1}) \circ \ldots \circ D(1^n, s)(\overline{p(n)})$$

Where we assume that:

- We use the notations $\overline{0}, \overline{1}, \ldots, \overline{p(n)}$ to mean the binary string of length $l(n)$ that represents the numbers. so $\overline{1}$ means $0^{l(n)-1} \circ 1$.
- The output of $D(1^n, s)$ represents a function in $\{\mathcal{F}_n\}$. We use this notation as the function itself. Thus we can write $D(1^n, s)(\overline{1})$
- Note that we relay on the fact that $2^{l(n)} > (p(n))$. Otherwise the above definition of g is invalid. Make sure you understand why .....
- To ease the writing, define $r(n) = m(n) \cdot (p(n) + 1)$ - the size of g's output.

Claim 0.7. *The above g is PRG.*

Proof of Claim 0.7. Clearly g enlarges the input size (even if $m(n) = 1$). So the major part of the proof is to show that $g(U_{p(n)})$ can't be (efficiently) distinguish from $U_{r(n)}$. So assume $A(1^n, y \in \{0,1\}^{r(n)})$ is a distinguisher, we'll prove it succeeds with negligible probability. Consider the following oracle-aided algorithm $\widetilde{A}(1^n)$, meant to distinguish $\{\mathcal{F}_n\}$ from $\Pi_{l(n),m(n)}$:

Algorithm 0.8 ($\widetilde{A}$: break $\mathcal{F}_n$)**.**

input*:* 1^n

- *for* $(i = 0$ *to* $p(n))$ *do:*
- $\quad y_i := Orcl(\overline{i})$
- *set* $y := y_0 \circ y_1 \circ \ldots \circ y_{p(n)}$
- *if*$(A(y) = 1)$
- $\quad$ *return 1*
- *else*
- $\quad$ *return 0*

So in case $\widetilde{A}$ is working with an oracle from $\{\mathcal{F}_n\}$, what it does is randomly choose an input to g (which as we explained before is the same as choosing a function from $\{\mathcal{F}_n\}$), compute g on it, and let A decide whether it came from $g(U_{p(n)})$ or from $U_{r(n)}$. $\widetilde{A}$ will succeed iff A will succeed, and the distribution is the same. Formally:

$$\begin{aligned}\mathsf{Pr}_{f \leftarrow \mathcal{F}_n}[\widetilde{A}^f = 1] &= \mathsf{Pr}_{f \leftarrow \mathcal{F}_n}[A(f(\overline{0}) \circ f(\overline{1}) \ldots f(\overline{p(n)})) = 1] \\ &= \mathsf{Pr}_{s \leftarrow U_{p(n)}}[A(D(1^n, s)(\overline{0}) \circ D(1^n, s)(\overline{1}) \ldots D(1^n, s)(\overline{p(n)})) = 1] \\ &= \mathsf{Pr}_{s \leftarrow U_{p(n)}}[A(g(s)) = 1] \end{aligned} \tag{11}$$

And about working in $\Pi_{l(n),m(n)}$:

$$\begin{aligned}\mathsf{Pr}_{f \leftarrow \Pi_{l(n),m(n)}}[\widetilde{A}^f = 1] &= \mathsf{Pr}_{f \leftarrow \Pi_{l(n),m(n)}}[A(f(\overline{0}) \circ f(\overline{1}) \ldots f(\overline{p(n)})) = 1] \\ &= \mathsf{Pr}_{r \leftarrow U_{r(n)}}[A(r) = 1] \end{aligned} \tag{12}$$

Combining (11), (12), and the fact that $\{\mathcal{F}_n\}$ is PRF we get:

$$\begin{aligned}\left|\mathsf{Pr}_{r\leftarrow U_{r(n)}}[A(r)=1]-\mathsf{Pr}_{s\leftarrow U_{p(n)}}[A(g(s))=1]\right| &= \left|\mathsf{Pr}_{f\leftarrow \Pi_{l(n),m(n)}}[\widetilde{A}^f=1]-\mathsf{Pr}_{f\leftarrow \mathcal{F}_n}[\widetilde{A}^f=1]\right| \\ &< neg(n)\end{aligned}$$

Since A was arbitrary distinguisher, the proof is done. □

Remark 0.9. *Note that our PRG g is defined only for strings of length $\{p(n) \mid n \in \mathbb{N}\}$. If we want to define it also for other length we can use the following simple fact: If $f : \{0,1\}^n \to \{0,1\}^{l(n)}$, is a PRG, then also the following function is a PRG: $f' : \{0,1\}^{n+1} \to \{0,1\}^{l(n)+1}$, defined as $f'(s \circ a) = f(s) \circ a$. $(a \in \{0,1\})$*

Exe 4. Constructing pairwise-independent function family. (10 points) Recall that a function family $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is *pairwise independent*, if for any for any $x \neq x' \in \{0,1\}^n$ it holds that

$$\mathsf{Pr}_{h \leftarrow \mathcal{H}}[h(x) = h(x')] = 2^{-m}$$

(That is, the probability that two fixed points in the domain collide under h is exactly the same as if h were a truly random function from $\{0,1\}^n$ to $\{0,1\}^m$.)

There are many combinatorial constructions of efficient ensembles of pairwise independent hash functions with short description, in the following we consider one such a family.

Let $A_{m \times n}$ be the set of all $m \times n$ binary matrices. Show that the family $\mathcal{H} := \{h_{A,b} \colon A \in A_{n \times m}, b \in \{0,1\}^m\}$, where $h_{A,b}(x) \equiv Ax + b \bmod 2$, is pairwise independent.

solution 4 In the following solution, all arithmetic operation are taken mod 2. Suppose $x_1 \neq x_2 \in \{0,1\}^n$. The following holds for any $h \in \mathcal{H}$:

$$h(x_1) = h(x_2) \quad \Leftrightarrow \quad A \cdot x_1 + b = A \cdot x_2 + b \quad \Leftrightarrow \quad A \cdot x_1 = A \cdot x_2 \quad \Leftrightarrow \quad A \cdot (x_1 - x_2) = \overrightarrow{0}$$

Hence we get that:

$$\mathsf{Pr}_{h \leftarrow \mathcal{H}}[h(x_1) = h(x_2)] = \mathsf{Pr}_{A \leftarrow A_{m \cdot n}}[A(x_1 - x_2) = \overrightarrow{0}] \tag{13}$$

Define $x' = x_1 - x_2$. Since $x_1 \neq x_2$, we get that $x' \neq \overrightarrow{0}$, hence there is an index j, such that $x'[j] = 1$.
Consider an arbitrary index in the vector $A \cdot x'$ - call it i, $1 \leq i \leq n$. We have:

$$\begin{aligned}
[A \cdot x']_i &= \sum_{k=1}^{n} A[i,k] \cdot x'[k] \\
&= A[i,j] \cdot x'[j] + \sum_{k=1, k \neq j}^{n} A[i,k] \cdot x'[k] \\
&= A[i,j] + \sum_{k=1, k \neq j}^{n} A[i,k] \cdot x'[k]
\end{aligned}$$

Denote the value of the last sum as v. No matter what v is we get:

$$\mathsf{Pr}_{A \leftarrow A_{m \cdot n}}[[A \cdot x']_i = 0] = \mathsf{Pr}_{A \leftarrow A_{m \cdot n}}[A[i,j] = v] = \frac{1}{2}$$

i was an arbitrary index in the vector $A \cdot x'$, hance we have:

$$\mathsf{Pr}_{A \leftarrow A_{m \cdot n}}[A \cdot x' = \overrightarrow{0}] = (\frac{1}{2})^m$$

Combining it with (13) we get the desired result:

$$\mathsf{Pr}_{h \leftarrow \mathcal{H}}[h(x_1) = h(x_2)] = (\frac{1}{2})^m$$

Exe 5, PRF domain extension. Let $\mathcal{F} = \{\mathcal{F}_n = \{f\colon \{0,1\}^n \mapsto \{0,1\}^n\}\}_{n\in\mathbb{N}}$ be a PRF, and let $\mathcal{H} = \{\mathcal{H}_n = \{h\colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}\}_{n\in\mathbb{N}}$ be an efficient pairwise-independent function family.[1] We would like to prove that the function family ensemble $\mathcal{F} \circ \mathcal{H} = \{\mathcal{F}_n \circ \mathcal{H}_n = \{f \circ h\colon f \in \mathcal{F}_n, h \in \mathcal{H}_n\}\}_{n\in\mathbb{N}}$ is a PRF mapping strings of length $2n$ to string of length n.[2]

(10 points) Prove that function family ensemble $\{\Pi_n \circ \mathcal{H}_n\}_{n\in\mathbb{N}}$ is computationally (actually, also statistically) indistinguishable from $\{\Pi_{2n,n}\}$.

(10 points) Use the above to prove that $\mathcal{F} \circ \mathcal{H}$ is a PRF.

Exe 5, PRF domain extension. Let $\mathcal{F} = \{\mathcal{F}_n = \{f\colon \{0,1\}^n \mapsto \{0,1\}^n\}\}_{n\in\mathbb{N}}$ be a PRF, and let $\mathcal{H} = \{\mathcal{H}_n = \{h\colon \{0,1\}^{2n} \mapsto \{0,1\}^n\}\}_{n\in\mathbb{N}}$ be an efficient pairwise-independent function family.[3] We would like to prove that the function family ensemble $\mathcal{F} \circ \mathcal{H} = \{\mathcal{F}_n \circ \mathcal{H}_n = \{f \circ h\colon f \in \mathcal{F}_n, h \in \mathcal{H}_n\}\}_{n\in\mathbb{N}}$ is a PRF mapping strings of length $2n$ to string of length n.[4]

(10 points) Prove that function family ensemble $\{\Pi_n \circ \mathcal{H}_n\}_{n\in\mathbb{N}}$ is computationally (actually, also statistically) indistinguishable from $\{\Pi_{2n,n}\}$.

(10 points) Use the above to prove that $\mathcal{F} \circ \mathcal{H}$ is a PRF.

Solution 5:a

Claim 0.10. *Suppose $x_1, x_2, \ldots, x_k \in \{0,1\}^{2n}$ are k different values. Then:*

$$\mathsf{Pr}_{h\leftarrow\mathcal{H}_n}[h(x_1), \ldots, h(x_k) \text{ are } k \text{ different values }] \;\geq\; 1 - \frac{k^2}{2^n}$$

Proof of Claim 0.10.

$$\begin{aligned}
\mathsf{Pr}_{h\leftarrow\mathcal{H}_n}[h(x_1), \ldots, h(x_k) \text{ are } k \text{ different values }] &= \mathsf{Pr}_{h\leftarrow\mathcal{H}_n}[\cap_{i=1,j=1}^{k} h(x_i) \neq h(x_j)] \\
&= 1 \;-\; \mathsf{Pr}_{h\leftarrow\mathcal{H}_n}[\overline{\cap_{i=1,j=1}^{k} h(x_i) \neq h(x_j)}] \\
&\geq 1 - \sum_{i=1,j=1}^{k} \mathsf{Pr}_{h\leftarrow\mathcal{H}_n}[h(x_i) = h(x_j)] \\
&= 1 - \frac{k^2}{2^n}
\end{aligned}$$

□

Define the following subset of $\mathcal{H}_n$:

[1] Namely, the family $\mathcal{H}_n$, for each $n \in \mathbb{N}$, is pairwise independent.
[2] The symbol $\circ$ stands for function concatenation, e.g., $f \circ h(x) = f(h(x))$.
[3] Namely, the family $\mathcal{H}_n$, for each $n \in \mathbb{N}$, is pairwise independent.
[4] The symbol $\circ$ stands for function concatenation, e.g., $f \circ h(x) = f(h(x))$.

Definition 0.11. *Suppose $x_1, \ldots, x_l$ are different values in $\{0,1\}^{2n}$. Then denote:*

$$\mathcal{H}_n^{\{x_1,\ldots,x_l\}} \;=\; \{h \in \mathcal{H}_n \;\mid \{h(x_1), h(x_2), \ldots, h(x_l)\} \textit{ are different}\}$$

So the definition above, looks only at those function h, that give different values to those x_i. The following claim says that if we take an additional x, with a high probability its image under an arbitrary h will differ from previous values chosen by h. That is true even if sample h from $\mathcal{H}_n^{\{x_1,\ldots,x_l\}}$

Claim 0.12. *Suppose $x_1, \ldots, x_l$ are different values in $\{0,1\}^{2n}$. Then:*

$$\mathsf{Pr}_{h \leftarrow \mathcal{H}_n^{\{x_1,\ldots,x_{l-1}\}}}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \textit{ are different}] \geq 1 - \frac{l^2}{2^n}$$

Proof of Claim 0.12. The following holds:

$$\begin{aligned}
&\mathsf{Pr}_{h \leftarrow \mathcal{H}_n^{\{x_1,\ldots,x_{l-1}\}}}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \text{ are different}] \\
&= \mathsf{Pr}_{h \leftarrow \mathcal{H}_n}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \text{ are different} \mid \{h(x_1), h(x_2), \ldots, h(x_{l-1})\} \text{ are different}] \\
&= (\mathsf{Pr}_{h \leftarrow \mathcal{H}_n}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \text{ different}]) \;/\; (\mathsf{Pr}_{h \leftarrow \mathcal{H}_n}[\{h(x_1), h(x_2), \ldots, h(x_{l-1})\} \text{ different}]) \\
&\geq \mathsf{Pr}_{h \leftarrow \mathcal{H}_n}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \text{ different}] \\
&\geq 1 - \frac{l^2}{2^n}
\end{aligned}$$

The second equality is due to conditional probability definition, and the last step is due Claim 0.10. □

Now we are ready for the main claim we need, that states that for every distinguisher D, the probability to get a collision in h (when it uses an oracle from $\Pi_n \circ \mathcal{H}_n$) is very low. For that we first need to define several notations. So we assume now that we have an all powerful algorithm $D^O(1^n)$. During its run D makes several oracle calls. It passes values to the oracle, and the oracle calculate the returned value in 2 phases (first h, than π). Those values are actually random variables depend on the randomness we used to choose $\pi \circ h$, and (maybe) the randomness that D uses. Denote those random variables as:

- X_i - The i-th (different) value that D passes to the oracle.
- T_i - The i-th intermediate value that the oracle calculate, that is: $h(X_i)$
- Y_i - The response of the oracle to X_i
- Also denote by $x_i,\ t_i = h(x_i),\ y_i$ the values of those random variables in a specific run

Here is out main claim:

Claim 0.13. *Suppose $D^O(1^n)$ is running with an oracle randomly chosen from $\{\Pi_n \circ \mathcal{H}_n\}$. Then*

$$\mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[\{T_1, T_2, \ldots, T_k\} \text{ are all different}] \geq 1 - \frac{k^3}{2^n}$$

Proof of Claim 0.13. As a first step lets try to (lower) bound the following:

$$\mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_l \text{ different from } \{T_1, T_2, \ldots, T_{l-1}\} \mid \{T_1, T_2, \ldots, T_{l-1}\} \text{ are different}]$$

So now D has come to the stage that it choose x_l. We assume that $\{h(x_1), \ldots, h(x_{l-1})\}$ are different values. We want to calculate the probability that $h(x_l)$ will also be different from all previous $h(x_i)$s, or actually to give a lower bound for that. Since after the $h(x_i)$ is calculated, the oracle apply to it a random function π, the values that D saw $y_1, y_2, \ldots, y_{l-1}$, look to it as pure random values. The key point here is to realize that no matter what D does, the only information it has, is that the h it is working with, haven't collided on $\{x_1, \ldots, x_{l-1}\}$. Hence information theoretically, it can't get a better result than if it would have worked with h sampled from $\mathcal{H}_n^{\{x_1,x_2,\ldots,x_{l-1}\}}$. So we get the following lower bound:

$$\begin{aligned}
&\mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_l \text{ different from } \{T_1, T_2, \ldots, T_{l-1}\} \mid \{T_1, T_2, \ldots, T_{l-1}\} \text{ are different}] \\
&\quad \geq \mathsf{Pr}_{h\leftarrow \mathcal{H}_n^{\{x_1,\ldots,x_{l-1}\}}}[\{h(x_1), h(x_2), \ldots, h(x_l)\} \text{ are different}] \\
&\quad \geq 1 - \frac{l^2}{2^n} \qquad (14)
\end{aligned}$$

The last inequality is due to Claim 0.12.
Now we are able to give a lower bound for the desired event. Using conditional probability we have:

$$\begin{aligned}
&\mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[\{T_1, T_2, \ldots, T_k\} \text{ are all different}] = \\
&= \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_k \text{ different from } \{T_1, T_2, \ldots, T_{k-1}\} \mid \{T_1, T_2, \ldots, T_{k-1}\} \text{ are different}] * \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[\{T_1, T_2, \ldots, T_{k-1}\} \text{ are different}] = \\
&= \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_k \text{ different from } \{T_1, T_2, \ldots, T_{k-1}\} \mid \{T_1, T_2, \ldots, T_{k-1}\} \text{ are different}] * \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_{k-1} \text{ different from } \{T_1, T_2, \ldots, T_{k-2}\} \mid \{T_1, T_2, \ldots, T_{k-2}\} \text{ are different}] * \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[\{T_1, T_2, \ldots, T_{k-2}\} \text{ are different}] = \\
&= \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_k \text{ different from } \{T_1, T_2, \ldots, T_{k-1}\} \mid \{T_1, T_2, \ldots, T_{k-1}\} \text{ are different}] * \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_{k-1} \text{ different from } \{T_1, T_2, \ldots, T_{k-2}\} \mid \{T_1, T_2, \ldots, T_{k-2}\} \text{ are different}] * \\
&\quad \ldots \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_3 \text{ different from } \{T_1, T_2\} \mid \{T_1, T_2\} \text{ are different}] * \\
&\quad \mathsf{Pr}_{\pi\circ h\leftarrow \Pi_n\circ\mathcal{H}_n}[T_1 \neq T_2]
\end{aligned}$$

Now applying (14) we get that:

$$\begin{aligned}\mathsf{Pr}_{\pi\circ h\leftarrow\Pi_n\circ\mathcal{H}_n}[\{T_1,T_2,\ldots,T_k\} \text{ are all different}] &\geq (1-\frac{k^2}{2^n})\cdot(1-\frac{(k-1)^2}{2^n})\cdot\ldots\cdot(1-\frac{(1)^2}{2^n})\\ &\geq (1-\frac{k^2}{2^n})^k\\ &\geq 1-\frac{k^3}{2^n}\end{aligned}$$

Where the last step is due to Bernuli's inequality. □

Now we can apply the coupling technique and prove that $\{\Pi_n \circ \mathcal{H}_n\}$, is statistically indistinguishable from $\{\Pi_{2n,n}\}$. So consider an all powerful distinguisher $D^O(1^n)$ that is restricted to polynomial number of queries (denote this number by k), aimed to distinguish between $\{\Pi_n \circ \mathcal{H}_n\}$ and $\{\Pi_{2n,n}\}$.
For a function $g : \{0,1\}^{2n} \to \{0,1\}^n$, denote by $V(D^g(1^n))$, the view of D when it run using g. We'll see that:

$$SD\ (\ V(D^{\Pi_n\circ\mathcal{H}_n}(1^n))\ ,\ V(D^{\Pi_{2n,n}}(1^n)\ ) = neg(n)$$

And since the output of D is a part of the view we'll get the desired result that:

$$\left|\ \mathsf{Pr}_{\pi\circ h\leftarrow\Pi_n\circ\mathcal{H}_n}[D^{\pi\circ h}(1^n)=1]\ -\ \mathsf{Pr}_{\pi\leftarrow\Pi_{2n,n}}[D^{\pi}(1^n)=1]\ \right| = neg(n)$$

Let define 2 (dependent) random variables V_1, V_2 - as 2 views of D. We simulate D twice (in parallel), and we let V_1 be the view of simulation S_1, and V_2 the view of S_2. Here is the definition of S_1, S_2:
At the beginning we randomly choose a function $h \leftarrow \mathcal{H}_n$. Than we start to simulate D. When there is an oracle call for a new value x_l we do the following:

1. Randomly choose a value from $\{0,1\}^n$ (denote it as y_l), and return this value to S_1.
2. If $h(x_l)$ is different from all $h(x_i)$, for all $i \leq l$, we return y_l also to S_2. Otherwise, if $h(x_l) = h(x_j)$ we return the same y_j we returned to that x_j.

Obviously we have:

- The distribution of the view V_1, is the same distribution as of $V(D^{\Pi_{2n,n}}(1^n))$.
- The distribution of the view V_2, is the same distribution as of $V(D^{\Pi_n\circ\mathcal{H}_n}(1^n))$.

Now applying the coupling lemma (sent by Iftach) we get:

$$SD\ (\ V(D^{\Pi_n\circ\mathcal{H}_n}(1^n))\ ,\ V(D^{\Pi_{2n,n}}(1^n)\ ) \leq \mathsf{Pr}(V_1 \neq V_2)$$

From the definition of V_i, it's obvious that if h never collide on the different x_i then $V_1 = V_2$ we conclude that (using Claim 0.13) :

$$\mathsf{Pr}(V_1 \neq V_2) \leq \frac{k^3}{2^n} \leq neg(n)$$

solution 5:b

Claim 0.14. *The following 2 ensembles are computationally indistinguishable:* $\{\Pi_n \circ \mathcal{H}_n\}$, $\{\mathcal{F}_n \circ \mathcal{H}_n\}$

Proof of Claim 0.14. Let $D^O(1^{2n})$ be an oracle-aided PPT distinguisher, aimed to distinguish between $\{\Pi_n \circ \mathcal{H}_n\}$ and $\{\mathcal{F}_n \circ \mathcal{H}_n\}$. Consider the following oracle-aided PPT distinguisher, aimed to distinguish between $\{\Pi_n\}$ and $\{\mathcal{F}_n\}$.

Algorithm 0.15 ($\widetilde{D}$: break $\mathcal{F}_n$)**.**

input*:* 1^n

oracle*:* $O : \{0,1\}^n \to \{0,1\}^n$

- *let* $h \leftarrow \mathcal{H}_n$
- *create the following oracle:* $O_D : \{0,1\}^{2n} \to \{0,1\}^n \qquad O_D = O \circ h$
- *return* $D^{O_D}(1^{2n})$

We have:

$$\mathsf{Pr}_{\pi \leftarrow \Pi_n}[\widetilde{D}^{\pi}(1^n) = 1] = \mathsf{Pr}_{\pi \circ h \leftarrow \Pi_n \circ \mathcal{H}_n}[D^{\pi \circ h}(1^{2n}) = 1] \tag{15}$$

The same holds for $\mathcal{F}_n \circ \mathcal{H}_n$:

$$\mathsf{Pr}_{f \leftarrow \mathcal{F}_n}[\widetilde{D}^{f}(1^n) = 1] = \mathsf{Pr}_{f \circ h \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}[D^{f \circ h}(1^{2n}) = 1] \tag{16}$$

Combine (15) with (16), and the fact that $\mathcal{F}_n$ is PRF we get:

$$\begin{aligned} &\left|\mathsf{Pr}_{f \circ h \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}[D^{f \circ h}(1^{2n}) = 1] - \mathsf{Pr}_{\pi \circ h \leftarrow \Pi_n \circ \mathcal{H}_n}[D^{\pi \circ h}(1^{2n}) = 1]\right| \\ &= \left|\mathsf{Pr}_{f \leftarrow \mathcal{F}_n}[\widetilde{D}^{f}(1^n) = 1] - \mathsf{Pr}_{\pi \leftarrow \Pi_n}[\widetilde{D}^{\pi}(1^n) = 1]\right| \;\leq\; neg(n) \end{aligned}$$

□

Combining the last claim with 5.a we get that $\{\mathcal{F}_n \circ \mathcal{H}_n\}$ is computationally indistinguishable from $\{\Pi_{2n,n}\}$. Hence $\{\mathcal{F}_n \circ \mathcal{H}_n\}$ is a PRF